

Rukovanje događajima

U programiranju, događaj je nešto što nastaje u okviru aplikacijskog domena: recimo, klik miša, pritisak na dugme na tastaturi ili promene u samom grafičkom interefejsu (GUI). U Pythonu, koristeći Tkinter modul, možemo rukovati ovim događajima povezujući ih sa listom predefinisanih ili korisnički definisanih funkcija, što nam omogućava pokretanje željenog koda nakon pobude tog događaja. Događaji u GUI programiranju predstavljaju osnovnu celinu i aspekt kome treba posvetiti posebnu pažnju prilikom osmišljavanja aplikacije i pisanja samog koda.

Osnovna sintaksa prilikom povezivanja događaja u Tkinteru

Kako bismo povezali događaj sa željenom funkcijom, moramo koristiti `bind()` metodu, koja je deo svih vidžeta. Ova metoda sadrži dva argumenta:

- `event` – tip tekst (string), koji predstavlja sve detalje koji se tiču datog događaja; prosleđuje se u obliku `<modifier-type-detail>`; jedini obavezni deo ovog string argumenta je sekcija `"type"` – što predstavlja tip događaja koji se osluškuje; ostale dve sekcije, `modifier` i `detail`, koriste se za prosleđivanje dodatnih informacija, koje nisu neophodne za uspešno osluškivanje događaja i mogu se izostaviti;
- `handler` – ime funkcije koja će se **pozvati** pri nastanku željenog događaja; reč je samo o imenu funkcije, što znači da ne **može** imati i argumente.

Funkciji koja se prosleđuje kao rukovalac događaja se prosleđuje objekat događaja koji podrazumeva detalje o događaju koji je nastao.

Pogledajmo na primeru kako izgleda dodeljivanje rukovaoca događaju klika mišem na labelu koji će ispisivati poruku na ekranu prilikom pobude:

Primer događaja klik mišem

```
from tkinter import *
def mouse_click( event ):
    print( "Click event" )
window = Tk()
label = Label( window, text="Click here" )
label.pack()
label.bind( "<Button>", mouse_click )
window.mainloop()
```

Ovaj kod će prvo kreirati prozor sa jednom labelom i njenim tekstom `Click here`. Nakon klika (bez obzira na to da li je reč o levom ili desnom kliku ili točkiću), `mouse_click()` funkcija će se pozvati i u konzoli ispisati: `Click event`.

Izmenite kod tako da kada se klikne na dugme se ispisuje vrednost 5.

Tipovi događaja (event types)

U Tkinteru postoji nekoliko tipova događaja, od kojih su najbitniji navedeni u listi ispod:

- `KeyPress/Key` – aktivira se prilikom pritiska tastera na tastaturi;
- `KeyRelease` – aktivira se prilikom puštanja prethodno pritisnutog tastera;
- `Button` – aktivira se prilikom klika mišem;
- `ButtonRelease` – aktivira se prilikom puštanja prethodno pritisnutog klika;
- `Motion` – aktivira se prilikom prevlačenja kursora preko vidžeta;
- `FocusIn/Out` – aktivira se nakon što dati vidžet dobije fokus kroz korisničku interakciju (u našem primeru je to klik mišem na labelu);
- `Configure` – aktivira se kada se konfiguracija datog vidžeta promeni (primer: korisnik u realnom vremenu menja širinu datog vidžeta).

Modifikator događaja (event modifier)

Modifikator događaja može izmeniti okolnosti pri kojima se aktivira rukovalac događajem. Na primer, neki modifikatori zahtevaju pritiskanje dodatne kombinacije tastera kako bi se događaj pobudio.

Neki od uobičajenih modifikatora su sledeći:

- `Control` – zahteva pritiskanje i držanje `ctrl` tastera na tastaturi prilikom pobude događaja;
- `Alt` – zahteva pritiskanje i držanje `alt` tastera na tastaturi prilikom pobude događaja;
- `Shift` – zahteva pritiskanje i držanje `shift` tastera na tastaturi prilikom pobude događaja;
- `Double/Triple/Quadruple` – zahteva da se dati događaj desi dva/tri/četiri puta u trenutku (primer: dupli klik miša).

Detalji događaja (event details)

Sekcija sa detaljima string parametra koji se prosleđuje rukovaocu događaja nam omogućava preciznije opisivanje događaja. Primer: Samo pritiskom na određeni taster na tastaturi pobuditi određeni događaj.

- Kada koristimo `Button` ili `ButtonRelease`, kao detalj, možemo proslediti i numeričku vrednost od 1 do 5 koja predstavlja određeni taster na mišu koji će pobuditi događaj.
- Kada koristimo `Key`, `KeyPress` ili `KeyRelease` kao detalj, možemo proslediti i ASCII vrednost za željeni taster kojim će se pobuditi događaj.

Da bismo prethodni primer promenili tako da se događaj pobuđuje duplim klikom umesto jednim, moramo promeniti:

```
label.bind( "<Button>", mouse_click) u  
label.bind( "<Double-Button>", mouse_click)
```

U tom primeru smo promenili modifikator (`Double`) događaja. Ako bismo želeli da se događaj pobuđuje samo levim duplim klikom, tu liniju koda bismo zamenili sa:

```
label.bind( "<Double-Button-1>", mouse_click)
```

Ovde smo pored modifikatora iz prethodnog primera dodali i detalj 1, koji predstavlja levi klik mišem. Detalj 2 bi predstavljao desni klik.

Ako bismo ponovo pokrenuli taj primer, sa poslednjom izmenom – primetili bismo da se sada tekst *Click event* ispisuje samo ako se izvrši dupli klik levim tasterom, dok se u prvobitnom primeru ta linija ispisivala klikom na bilo koji taster.

Izmenite kod tako da kada se klikne na taster Enter se u komandnom prozoru ispisuje Enter clicked. Detalj za klik taj Enter-a je `<Return>`.

Objekat događaja (event object)

Objekat događaja koji se prosleđuje rukovaocu događaja u trenutku njegove pobude može se iskoristiti za prikupljanje i korišćenje bitnih informacija o događaju. Objekat događaja ima nekoliko korisnih svojstava/atributa:

- `widget` – vidžet koji je generisao događaj; reč je o Tkinter vidžet instanci objekta, a ne o samom imenu;
- `x,y` – trenutna pozicija miša u pikselima;
- `x_root, y_root` – trenutna relativna pozicija miša u odnosu na gornji levi ugao čitavog ekrana (ne aplikacijskog prozora), u pikselima;
- `char` – kod pritisnutog karaktera koji je pobudio događaj (postoji samo u događajima koji nastaju usled pritiska tastera na tastaturi);
- `keysym` – simbol tastera koji je pobudio događaj (postoji samo u događajima koji nastaju usled pritiska tastera na tastaturi);
- `keycode` – kod tastera koji je pobudio događaj (postoji samo u događajima koji nastaju usled pritiska tastera na tastaturi);
- `num` – broj tastera (postoji samo u događajima koji nastaju usled klika mišem);
- `width, height` – nove dimenzije vidžeta nad kojim se odigrao događaj;
- `type` – tip događaja (ceo broj).

Kao primer korišćenja svojstva događaja, korišćićemo `x,y`, `type` i ime widžeta za koji je događaj vezan.

Primer pristupa detaljima događaja sa konzolnim ispisom

```
from tkinter import *
def mouse_scroll(event):
    print("Mouse event at x={}, y={}".format(event.x, event.y))
    print("Event source widget: {}".format(event.widget))
    print("Event type: {}".format(event.type))

window = Tk()
label = Label(window, text="Click here")
label.pack()
label.bind( "<MouseWheel>", mouse_scroll)
window.mainloop()
```

Ispis:

```
Mouse event at x=45, y=17
Event source widget: .!label
Event type: 38
Mouse event at x=45, y=17
Event source widget: .!label
Event type: 38
```

Na ovom primeru smo uvideli kako se može raditi sa event događajem. Naime:

- za prikaz lokacije na kojoj se događaj izvršio, koristili smo attribute `x` i `y` (`x` i `y` komponente koordinate);
- za ime vidžeta nad kojim se događaj odvio, koristimo atribut `widget`;
- dok za tip događaja koristimo atribut `type`; ovim atributom dobijamo numeričku vrednost koja predstavlja jedan od predefinisanih događaja u Tkinteru; lista predefinisanih događaja se može naći u [izvornom kodu modula](#) (reč je o klasi `EventType`); u našem slučaju je to vrednost 38, koja predstavlja `MouseWheel` događaj (bilo da je to up ili down scroll).

Pitanje

Koji je obavezni deo string argumenta event metode?

- Modifikator
- **Tip**
- Detalj

Objašnjenje:

Tačan odgovor je tip. Tip je najbitniji deo string parametra, jer predstavlja tip događaja koji se osluškuje. Primer: Button, KeyPress itd.

Prva od promena koja se može primetiti je korišćenje `MouseWheel` – događaja korišćenja točkića na mišu, a ne klika, kao u prethodnim primerima. Analizirajmo liniju po liniju ispisa:

```
Mouse event at x=8, y=7
```

Ova linija nam pokazuje na kojim se koordinatama, u odnosu na glavni prozor aplikacije, događaj desio (u pikselima).

```
Event type: 38.
```

Ova linija nam pokazuje kog tipa je događaj. Ova vrednost je tipa ceo broj (integer). Pogledati [izvorni kod Tkinter modula](#) za značenje ove vrednosti, kao i za ostale vrednosti tipova događaja.

```
Event source widget: .!label
```

Poslednja linija konzolnog ispisa je sama po sebi jasna i govori o tome sa kog vidžeta događaj dolazi.

Protokoli

Do sada smo koristili `bind()` metodu nad vidžetom kako bismo povezali rukovalac događaja sa vidžetom. Postoji još jedan način kreiranja rukovaoca događaja: koristeći metodu `protocol()`. Razlika između ove dve metode je što se (`bind()`) poziva nad bilo kojim vidžetom, a `protocol()` se može pozvati samo nad korenskim vidžetom (koji predstavlja glavni prozor aplikacije), koji je u prethodnom primeru kreiran pod nazivom `window`. Jedan od najčešće korišćenih protokola je `WM_DELETE_WINDOW`, koji se koristi za definisanje akcije koja će se desiti nakon što korisnik zatvori prozor (klik na x u desnom uglu prozora). U sledećem primeru ćemo koristiti metodu `protocol()` za povezivanje događaja sa funkcijom `confirm_exit()`, koja će upitati korisnika da li želi da izađe iz aplikacije. Za potvrđan odgovor će se izvršiti metoda `.destroy()` naredbom: `window.destroy()`, kojom će se glavni prozor aplikacije ugasiti i time će se izaći iz programa. Ovu metodu je moguće pozvati nad bilo kojim drugim vidžetom koji želimo da uklonimo.

Primer `protocol()` metode

```
from tkinter import *
from tkinter import messagebox

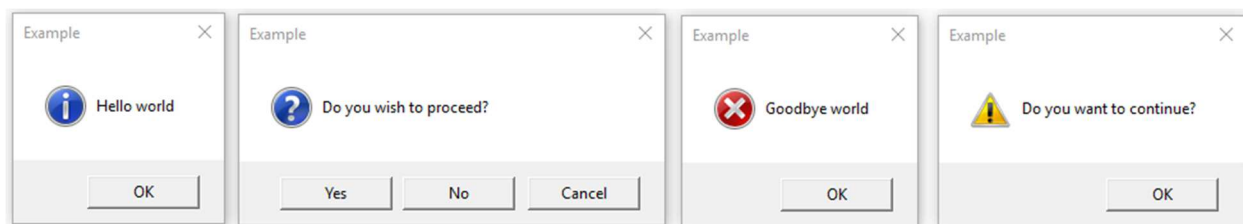
def confirm_exit():
    if messagebox.askokcancel("Exit", "Do you want to exit?"):
        window.destroy()

window = Tk()
Label = Label( window, text="Click here")
Label.pack()
window.protocol("WM_DELETE_WINDOW", confirm_exit)
window.mainloop()
```

Nakon ovog koda, Tkinter neće više sam automatski zatvoriti prozor, nego prvo pitati korisnika da potvrdi svoju odluku.

Neki od drugih korisnih protokola su `WM_TAKE_FOCUS` i `WM_SAVE_YOURSELF`. Za ostatak protokola, pogledati [ICCCM priručnik](#).

U ovom primeru smo takođe koristili i `messagebox` (okviri za poruke/dijalozi) – posebnu kontrolu za informisanje korisnika. Prilikom korišćenja mora se direktno uvesti iz `tkinter` biblioteke naredbom `from tkinter import messagebox`. Koristeći ovu metodu, korisniku možemo uputiti informaciju, upozorenje ili pitanje.



Slika 5.1. Različiti primeri izgleda *message boxa*¹

Detaljna lista `message boxova` se može podeliti u tri vrste: dijalozi informacija (information), dijalozi upozorenja (warning) i upitni (question) dijalozi.

- **Dijalozi informacija:**

- `tkinter.messagebox.showinfo(title=None, message=None, **options)`

- **Dijalozi upozorenja:**

- `tkinter.messagebox.showwarning(title=None, message=None, **options)`
- `tkinter.messagebox.showerror(title=None, message=None, **options)`

- **Upitni dijalozi:**

- `tkinter.messagebox.askquestion(title=None, message=None, **options)`
- `tkinter.messagebox.askokcancel(title=None, message=None, **options)`
- `tkinter.messagebox.askretrycancel(title=None, message=None, **options)`
- `tkinter.messagebox.askyesno(title=None, message=None, **options)`
- `tkinter.messagebox.askyesnocancel(title=None, message=None, **options)`

Ono što je zajedničko za sve ove pozive je to da se uvek može proslediti parametar `title`, koji označava naziv tog prozora, kao i argument `message`, koji označava poruku koja će pisati u dijalogu.

Izmeniti primer tako da se umesto upitnog dijaloga otvara dijalog upozorenja metode `showwarning` posle kog se prozor zatvara.

¹ <https://docs.python.org/3.9/library/tkinter.messagebox.html>

Rezime

- U programiranju, događaj je nešto što nastaje u okviru aplikacijskog domena: recimo, klik miša, pritisak na dugme na tastaturi ili promene u samom grafičkom interefejsu (GUI).
- Događaji u GUI programiranju predstavljaju osnovnu celinu i aspekt kome treba posvetiti posebnu pažnju prilikom osmišljavanja aplikacije i pisanja samog koda.
- Metoda `bind()` sadrži dva argumenta: event i handler.
- Modifikator događaja može izmeniti okolnosti pri kojima se aktivira rukovalac događajem. Na primer, neki modifikatori zahtevaju pritiskanje dodatne kombinacije tastera kako bi se događaj pobudio. Neki od uobičajenih modifikatora su: Control, Alt, Shift, Double/Triple/Quadruple.
- Sekcija sa detaljima string parametra koji se prosleđuje rukovaocu događaja omogućava preciznije opisivanje događaja.
- Objekat događaja koji se prosleđuje rukovaocu događaja u trenutku njegove pobude može se iskoristiti za prikupljanje i korišćenje bitnih informacija o događaju.
- Metoda `protocol()` se može pozvati jedino nad korenskim vidžetom, koji je u primeru kreiran pod nazivom window.
- Lista messageboxova se može podeliti u tri vrste: dijalog informacija (information message box), dijalozi upozorenja (warning message boxes) i upitni dijalozi (question message boxes).

